Tienda de comics JD

Juan Diego García Vargas

Diego Alejandro Rosazza Merida

Versión <x.x>

Cochabamba, <día> de <Mes> de <Año>

Registro Histórico de Cambios y Revisiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor Responsable** | **Aprobado** |
| 05/09/2018 | 0.0 | Presentación del tema | Juan Diego Garcia | 05/09/2018 |
| 05/09/2018 | 0.0 | Presentación de los Objetivos | Diego Rosazza | 05/09/2018 |
| 14/09/2018 | 0.3 | Presentación del Diagrama de Casos de Uso | Diego Rosazza | 14/09/2018 |
| 05/09/2018 | 0.0 | Presentación del Modelo de Datos | Juan Diego García | 05/09/2018 |
| 17/09/2018 | 0.1 | Presentación del Diagrama de Clases | Diego Rosazza | 14/09/2018 |
| 07/09/2018 | 0.0 | Subida Primeras relaciones funcionales | Juan Diego García | 07/09/2018 |
| 10/09/2018 | 0.1 | Creados métodos y entidades restantes | Diego Rosazza | 11/09/2018 |
| 11/09/2018 | 0.2 | Presentación del Primer Prototipo | Diego Rosazza | 11/09/2018 |
| 12/09/2018 | 0.3 | Reestructuración de todos los menús | Juan Diego García | 12/09/2018 |
| 17/09/2018 | 0.4 | Añadidos métodos faltantes | Juan Diego García | 17/09/2018 |
| <día/mes/año> | <x.x> | <Detalles del Cambio> | <Nombre> | <día/mes/año> |
| <día/mes/año> | <x.x> | <Detalles del Cambio> | <Nombre> | <día/mes/año> |
| 18/09/2018 | <x.x> | Presentación del Proyecto Final | Diego Rosazza | 18/09/2018 |

Tabla de Contenidos

1. Introducción 4

2. Análisis de Requerimientos 4

2.1 Requerimiento *«<Nombre del Requerimiento>»* 4

3. Objetivo General 4

4. Objetivos Específicos 4

5. Límites y Alcances 4

6. Marco Teórico 4

7. Diagramas de Casos de Uso 4

7.1 Caso de Uso «<Nombre del Caso de Uso>» 4

8. Diagrama de Clases 4

9. Pruebas y Validación 4

10. Conclusiones 4

11. Recomendaciones 4

12. Bibliografía 4

13. Anexos 4

13.1 Formularios 4

13.2 Autorizaciones 4

Informe Final del Proyecto

# Introducción

Los comics son una de las formas de narrar una historia más conocidos en estos días, los cuales suelen tratar las historias de héroes y grandes batallas, las cuales ocurren en nuestro mismo espacio y tiempo, generalmente reflejando la realidad social que se vive en una época, por ende, son complejos y variados.

Por ende, una tienda de comics debe ser capaz de mantener un registro completo y detallado de los comics, junto con la información necesaria para poder manejar una tienda de manera eficiente.

Es necesario crear una base de datos capaz de manejar el inventario y compra/venta de comics.

Cada comic tiene una serie diversa de datos a los que se les puede hacer referencia, muchos de estos compartidos por más de un comic, como ser los personajes que tiene, su distribuidora, los creadores del comic, los dibujantes del comic, su precio de compra, su precio de venta y si es de edición especial o cuenta con alguna nominación “especial”.

De los detalles de la distribuidora se tiene, los precios de venta, sus datos de venta y factura.

De los detalles de venta al cliente se tienen los detalles de ventas históricos más sus facturas.

Se debe poder ver, agregar, editar y eliminar cada una de estas clases, exceptuando las de venta y compra que no podrán ser editadas ni eliminadas

La base de datos es el gerente de la tienda, el cual recopila los datos de todos los movimientos hechos en cualquier sector y las características que engloba, teniendo los comics todos los datos para poder organizarlos y venderlos ya en la repisa, y siendo solo necesario que los empleados realicen el cobro de forma manual y directa según la información del comic.

Además, se tiene la opción de que tanto empleados como clientes puedan ver las características respecto a un comic, el cual no se extiende más allá de los precios de venta y no muestra los precios de compra, pero puede profundizar en otros datos relacionados

# Análisis de Requerimientos

En la base de datos a crearse, se tienen 2 usuarios distintos que pueden usar el sistema, uno es el gerente, el cual tiene control administrativo de toda la tienda ay cada uno de sus detalles, pudiendo modificar, eliminar, mostrar o actualizar la mayoría, exceptuando las de carácter financiero, con las que goza de una libertad limitada para manejar esos datos. El otro es el cliente, el cual puede ver todos los valores

## Gerente

### **Requerimiento “Manejar datos de Comics”**

* Poder crear, editar, listar y eliminar comics.
* Que no se repita el código de un comic.
* Ninguna

### **Requerimiento “Manejar datos de dibujantes”**

* Poder crear, editar, listar y eliminar dibujantes.
* Que no se repita el código de un dibujante.
* ninguna

### **Requerimiento “Manejar datos de distribuidora”**

* Poder crear, editar, listar y eliminar distribuidoras.
* Que no se repita el código de una distribuidora.
* ninguna

### **Requerimiento “Manejar datos de guionista”**

* Poder crear, editar, listar y eliminar guionistas.
* Que no se repita el código de un guionista.
* ninguna

### **Requerimiento “Manejar datos de Personajes”**

* Poder crear, editar, listar y eliminar Personajes.
* Que no se repita el código de un personaje.
* ninguna

### **Requerimiento “Manejar datos de Condición Especial”**

* Poder crear, editar, listar y eliminar Condiciones Especiales.
* Que no se repita el código de una Condición Especial.
* Ninguna

### **Requerimiento “Manejar datos de detalles de compra y venta”**

* Poder crear, editar, listar y eliminar detalles relacionados a la Compra/venta.
* Que no se repita el código de una Compra/Venta o que no exista una venta o comic asociado al detalle.
* Ninguna

### **Requerimiento “Manejar datos de compra y venta”**

* Poder crear, editar y listar Compras/ventas.
* Que no se repita el código de una Compra/Venta.
* ninguna

### **Requerimiento “Manejar datos de facturas”**

* Poder crear, editar, listar y anular detalles relacionados a las facturas de Compra/venta (Una factura anulada anula la compra/venta asociada).
* Que no se repita el código de una Compra/Venta.
* ninguna

### **Requerimiento “Realizar consultas de compras/ventas”**

* Realizar consultas según los datos deseados de compras o ventas.
* ninguna
* ninguna

### **Requerimiento “Realizar consultas de ganancias”**

* Poder ver las ganancias o pérdidas referentes a un comic y/o característica.
* ninguna
* ninguna

## Cliente

### **Requerimiento “Realizar consultas de la tienda”**

* Poder ver todos los detalles de la tienda no financieros que desee, como los comics disponibles, los guionistas, etc.
* ninguna
* ninguna

# Objetivo General

Se pretende tener un registro fiel y completo de una tienda de comics, capaz de ser usado por los clientes y el gerente de la tienda de forma independiente.

El gerente podrá controlar la totalidad de los datos internos en la tienda y además ver los que se encuentran en sistema y el cliente podrá informarse acerca de los productos que desea conseguir.

# Objetivos Específicos

[Se describen los resultados intermedios que se alcanzarán en el camino de lograr el Objetivo General. No están necesariamente relacionados con los requerimientos, sino más bien con la Metodología que se aplicará en el Desarrollo de la Solución]

Ser capaces de introducir, mostrar, modificar y eliminar datos en todas las secciones que lo requieran, teniendo de esta forma el esqueleto de la base de datos.

Crear métodos de consultas avanzados para obtener los datos deseados respecto al manejo económico de la tienda, es decir datos monetarios de ingresos y egresos en ciertos comics, personajes, guionistas, etc.

Tener un menú organizado orientado a la funcionalidad de los diferentes métodos, en los cuales se dividirán los métodos administrativos de los financieros y consultas específicas, el solo podrá ser usado por el gerente, el cual deberá introducir un código especial para acceder al mismo, teniendo el cliente un menú simplificado que le permitirá navegar por la base de datos relacionada a características y comics que pudiese desear.

# Límites y Alcances

[Una vez identificados los objetivos, es probable que por diferentes motivos (generalmente de disponibilidad de recursos) éstos no puedan ser alcanzados completamente. De acuerdo a la planificación realizada, este punto permite describir hasta dónde el presente trabajo pretende llegar]

La aplicación desarrollada solo podrá funcionar dentro de eclipse, sin tener ninguna interfaz gráfica distinta la barra de comandos, esto se debe al proceso de desarrollo de la base de datos, el cual para ser funcional demora mucho tiempo, por ende, no fuimos capaces de elaborar una interfaz grafica.

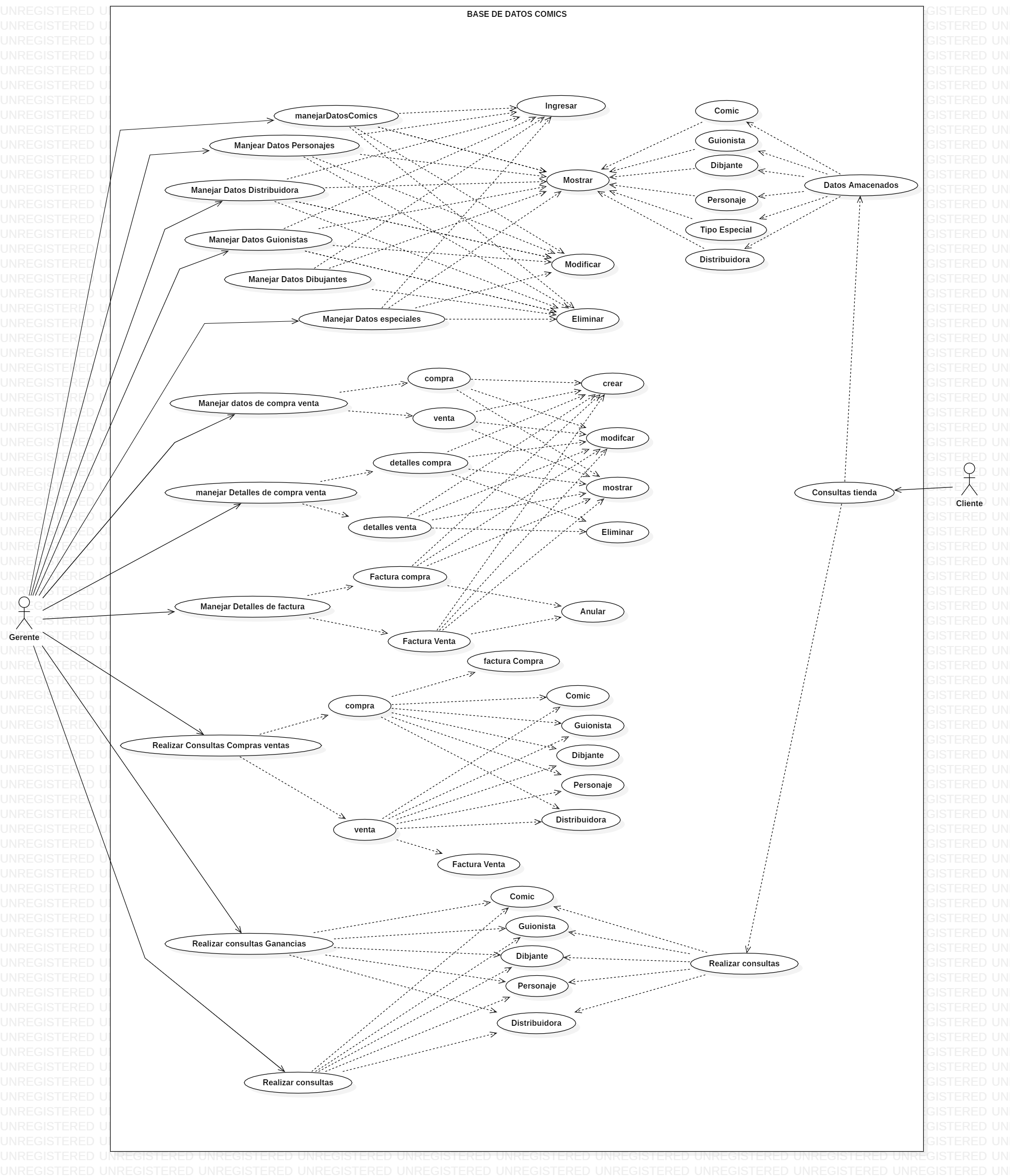
Debido a la forma en que se crearon las relaciones en los tipos especiales, si se quisiera tener un mismo comic con otra característica especial sería necesario tener otro código de barras para diferenciar los diferentes comics, con tal de tomarlos como entidades distintas. Cabe resaltar que la relación presente en el modelo de dbforge permite que un comic tenga varias características especiales, sin embargo, nos percatamos por el final que esto resultaría en que un comic, por especial que fuese, se vendería al mismo precio de uno regular, razón por la cual se creó esta limitante.

# Marco Teórico

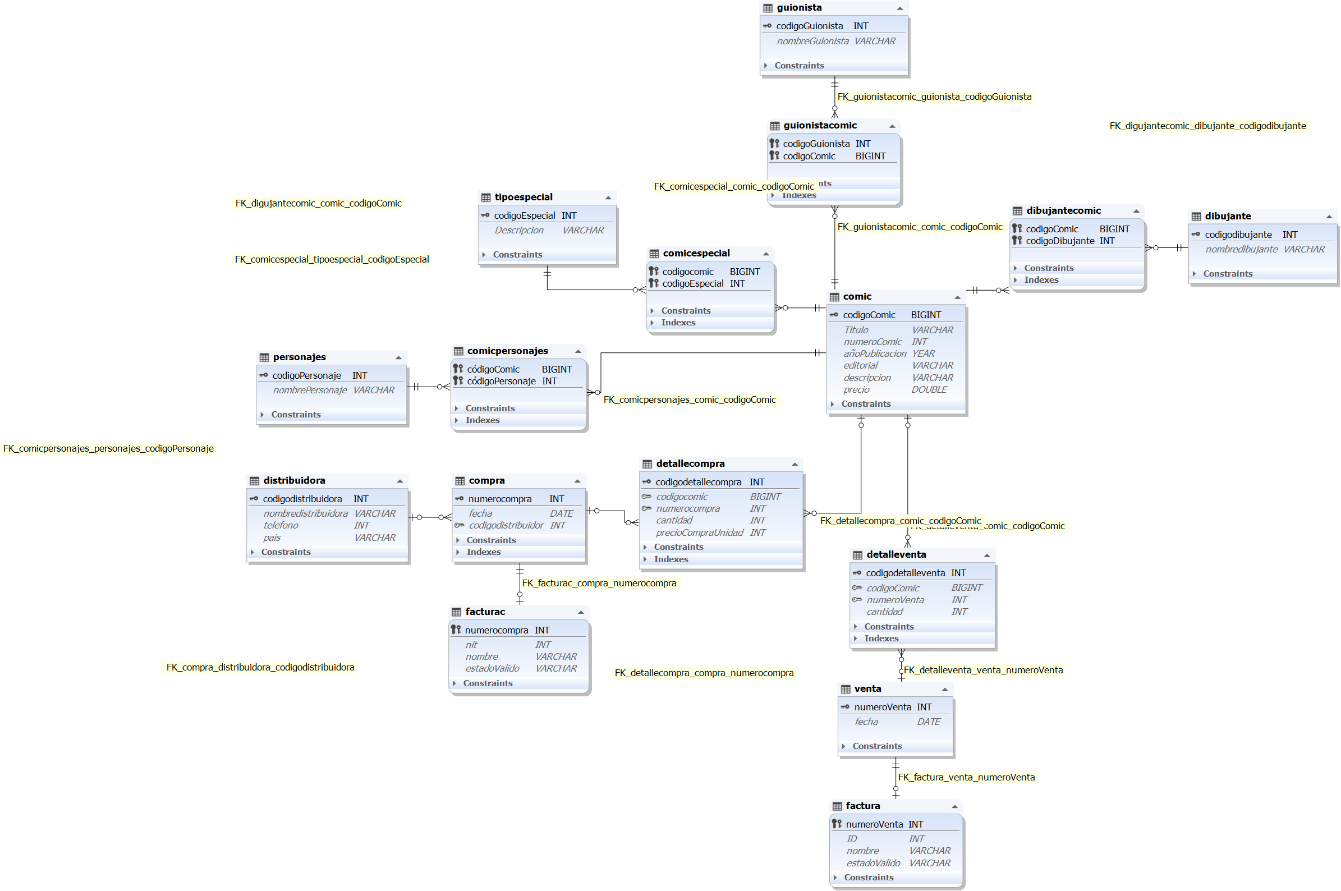
[Permite describir los conocimientos que fueron necesarios en el desarrollo de la solución, pero que no son parte del presente curso]

Fue necesario conocer el funcionamiento de java y el entorno de desarrollo eclipse, además del conocimiento interno de una tienda de comics, las características que trae un comic y las relaciones entre sus características.

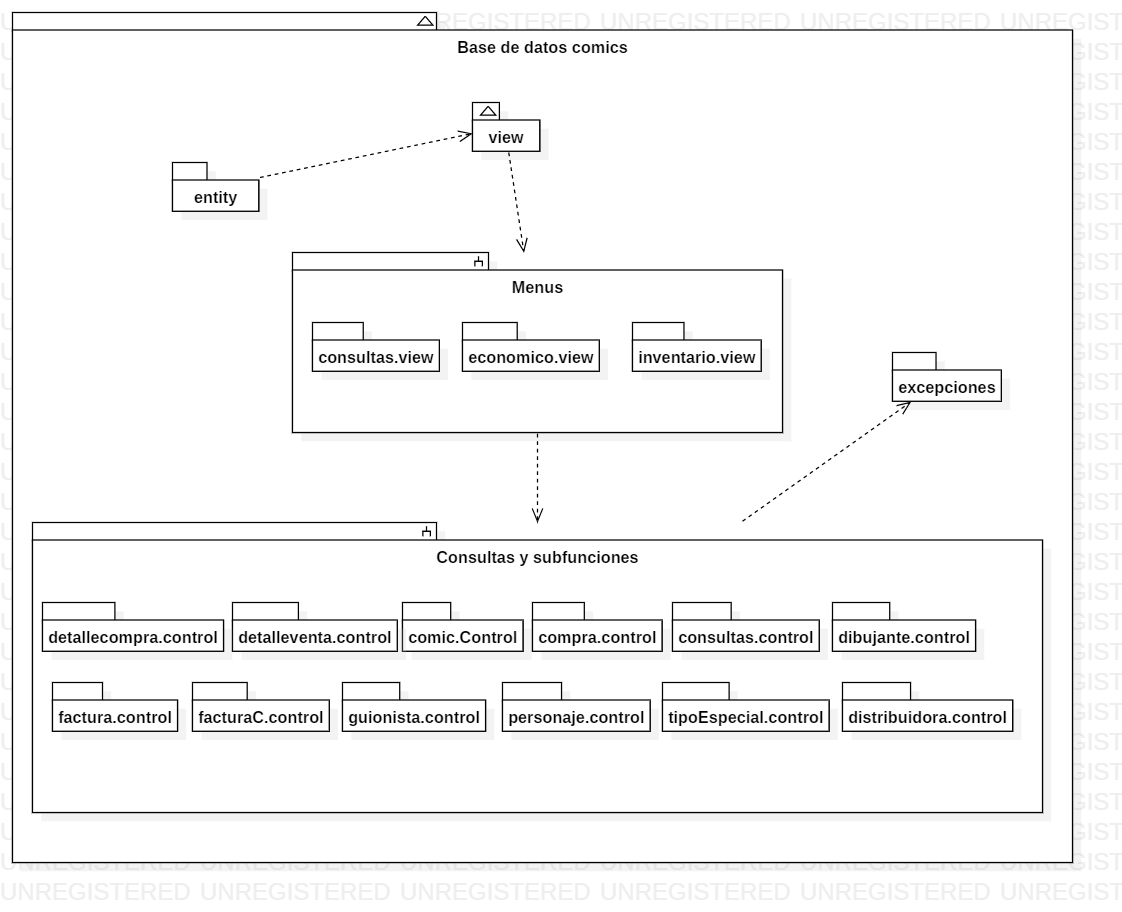
# Diagrama de Casos de Uso



# Modelo de Datos

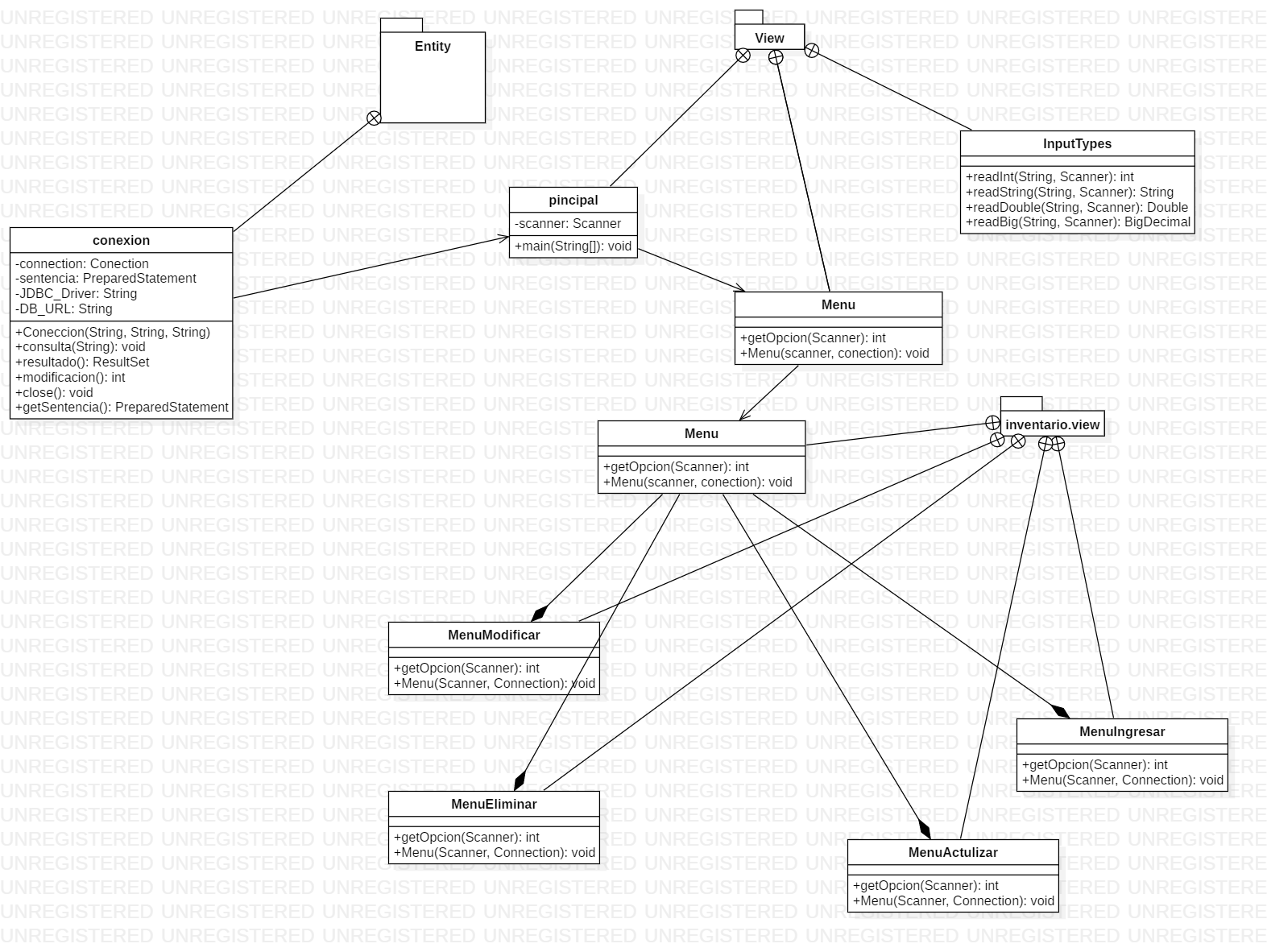


# Diagrama de Paquetes



[Describe cada uno de los objetos identificados para el desarrollo de su solución, caracterizados por sus atributos (propiedades), métodos (funcionalidades) y las relaciones que existen entre ellos. Normalmente existe una relación entre el Modelo de Datos y el Diagrama de Clases, pues es común tener una clase asociada a cada Tipo de Datos.]

Se crearon varios paquetes para poder identificar de mejor manera las funcionalidades de cada grupo identificad en el modelo de dbforge, es decir, las funcionalidades de cada tabla en la que se desean realizar consultas, crear datos, listar contenido, eliminar contenido y actualizar contenido, por ende, la mayoría de las funciones se repiten en cada paquete, cuya única diferencia recae en detalles en la orden de sql que se manda, para que esta se pueda usar en la tabla que desea, dicho esto, la relación entre clases depende de la relación entre paquetes que existe, ya que por ejemplo, el paquete inventario.view tiene internamente 5 clases, los cuales son menús, uno para elegir que acción de inventario se desea realizar y derivar al usuario al submenú correspondiente, el cual hace referencia a una clase correspondiente en otro paquete completamente distinto. Esto suena críptico, por ende se tiene la siguiente imagen para dar mas claridad a la forma en que todo esta estructurado por dentro.



Esta situación se repite para cada sub menú relacionado a una funcionalidad diferente, como el ver lo datos económicos o las consultas, razón por la cual no se realizara un diagrama completo de las clases, al ser estas repetitivas y dependientes de la relación entre paquetes.

Pero basta con saber que cada paquete tiene almacenada clases que hacen referencia directa a su nombre, es decir, todos los views tienen internamente menús para dirigir al usuario entre las opciones operacionales de los paquetes control, los cuales se encargan de realizar las consultas en sql, y dependiendo de que se pida de ellos, estos almacenaran clases distintas para cada consulta, dicho esto, todos los paquetes que tengan relaciones con inventario.view tienen internamente clases de eliminar, mostrar, crear y modificar, los paquetes relacionados con económico.view tienen métodos para crear, eliminar, mostrar y modificar, en los cuales las facturas reemplazan el eliminar por un anular, y los otros paquetes realizan consultas de ventas o compras de valores específicos, por ultimo, los paquetes relacionados a consultas.view manejan métodos para realizar consultas de cantidad y características deseadas de comics o características del mismo, a la vez de tener la ganancia neta según una característica deseada.

# Pruebas y Validación

[Se describen las pruebas que se aplicaron a la aplicación para validar el cumplimiento de los requerimientos, en el marco de los Límites y alcances planteados. Así como las pruebas que se aplicaron para validar las condiciones que se establecieron en el Análisis de Requerimientos. Si el resultado de una de las pruebas implicó realizar ajustes, éstos deben estar registrados en el Registro Histórico de Cambios y Revisiones. Esto implica también el cambio de versión en el documento (y la aplicación): si el cambio es menor, se incrementa la parte fraccionaria; si el cambio es mayor, se incrementa la parte entera y se reinicia la parte fraccionaria en cero. Todos estos cambios deben ser presentados y aprobados por el docente de la materia]

Al principio cuando creamos la base de datos en dbforge realizamos consultas internas para ver que las relaciones entre tablas funcionen de la forma correcta, posteriormente empezamos a replicar los métodos en java siguiendo el ejemplo dado por el docente referente a Tienda.

Para cada nueva consulta en java creada se realizó un a prueba de ejecución del comando creado desde java, por lo cual existe constancia de que el código creado funciona en su totalidad.

# Conclusiones

[Describe si se lograron los objetivos, los problemas que se encontraron y cómo se solucionaron]

# Recomendaciones

[En base a los objetivos, limitaciones y conclusiones, se describe cómo se puede continuar con el desarrollo de la aplicación; ya sea para mejorarla o para completarla]

# Bibliografía

[Enumerar los diferentes documentos que fueron consultados en todo el proceso del desarrollo del proyecto]

1.- Texto base de la materia

### 2.- [MariaDB Knowledge Base](https://mariadb.com/kb/es/)